REMARKS

Objection to the Abstract

Although Applicants respectfully note that there is no express requirement that the Abstract contain no numerical reference numbers, and Applicants respectfully note that many issued US patents contain numerical reference numbers (see as examples, US6,240,528 B1; US5,829,412; and US6,748,558B1), Applicants have amended the Abstract to remove the reference numbers in order to place this patent application in condition for allowance.

Rejections under 35 U.S.C. 103

Independent claims 1 and 14 were rejected under 35 U.S.C. 103(a) as being unpatentable over Leong (US Publication 2003/0078902) in view of InvFos (Infinitive Virtually Simultaneous File Opening System), and further in view of CapAfs (A Globally Accessible File System).

Claim Amendments:

The claims have been amended to clarify that the claimed invention is not specific to the JAVA programming language, but may be used with other programming languages.

Regarding claim 1:

Applicants respectfully note that Leong, InvFos, and CapAfs are all solving a different problem in a different manner than the claimed invention. Leong teaches running Java on a device on top of another programming

language, and accessing files through the interface between the two. CapAfs teaches a globally accessible file system with a uniform namespace that does not need any central authority to manage the namespace.

Applicants respectfully note that neither Leong nor CapAfs teaches "a method of emulating more simultaneously open files in the device than is permitted by an operating system of the device". InvFos does address the problem of not exceeding the allowable number of open files, but InvFos solves it in a very different way which does not require "emulating more simultaneously open files in the device than is permitted". InvFos teaches totaling the number of files which are already opened and the number of files to be opened to see if they exceed the limited number. If the limit is exceeded, InvFos teaches that the oldest file is closed and a file which is currently requested to be opened is opened. This is a very different solution than that taught by the claimed invention. InvFos does note teach "emulating more simultaneously open files in the device than is permitted", and neither do Leong and CapAfs.

In addition, unlike the present invention, InvFos is not trying to solve the problem in the more complex environment of having two programming languages. In InvFos, the environment is "a basic program". The present invention requires that there be two programming languages, which significantly increases the complexity of the problem. The solution taught by InvFos, merely closes the oldest file before a new file is opened. This would not solve the goal of the present invention, namely "emulating more simultaneously open files in the device than is permitted".

In addition, unlike the claimed invention, InvFos teaches only that opening a file, not reading or writing a file, triggers further steps. Thus, the

Examiner is incorrect to state that InvFos teaches the portions of claim 1 which read "if the file is open...." and "if the file is not open..." Nothing in InvFos takes a Read/Write request and turns it into a file open operation if the file is not open. And InvFos does not teach or even suggest "saving a file position for at least one open file, the file position designating where a next byte in the at least one open file would be accessed".

Applicants also respectfully note that the open, close, read, and write operation taught in CapAfs on pages 50-52 uses a file descriptor to recognize and associate a CapAfs file handle with an actual file. CapAfs is concerned with limiting access to only those files for which the requestor has permission. CapAfs is not at all concerned with "emulating more simultaneously open files in the device than is permitted". The CapAfs file descriptors have to do with checking for access rights and permissions; not "emulating more simultaneously open files in the device than is permitted" as required by the claimed invention.

The Examiner asserts that it would have been obvious to combine Leong, InvFos, and CapAfs because all three references are devoted to frequent file data manipulation. Applicants respectfully point out that virtually all computer-computer interactions involve frequent file data manipulation, as that is how computers transfer information. The Examiner is in essence saying that because the three prior art references deal with computers communicating, that that fact alone is enough suggestion to combine these three references. Applicants respectfully disagree. Applicants have found nothing to suggest combining these three references which are directed to solving very different problems in very different ways. Applicants does note that InvFos is directed to

solving the same problem as the present invention, however InvFos in fact teaches away from the present invention by solving the same problem in a very different way than the present invention. Thus, there is no motivation to combine these three references.

Remaining claims:

Regarding independent claim 14, The same arguments from claim 1 apply. The dependent claims are allowable for at least the same reasons as given for the independent claims.

Conclusion

Applicants believe the application is in condition for allowance which action is respectfully solicited. Please contact me if there are any issues regarding this communication or the current Application.

DOCKET NO. SC11854TS

Respectfully submitted,

SEND CORRESPONDENCE TO:

Freescale Semiconductor, Inc.
Law Department

Customer Number: 23125

By: Susan C. Hill

Susan C. Hill
Attorney of Record
Reg. No.: 35,896
Telephone: (512) 996-6849
Fax No.: (512) 996-6854
Email: